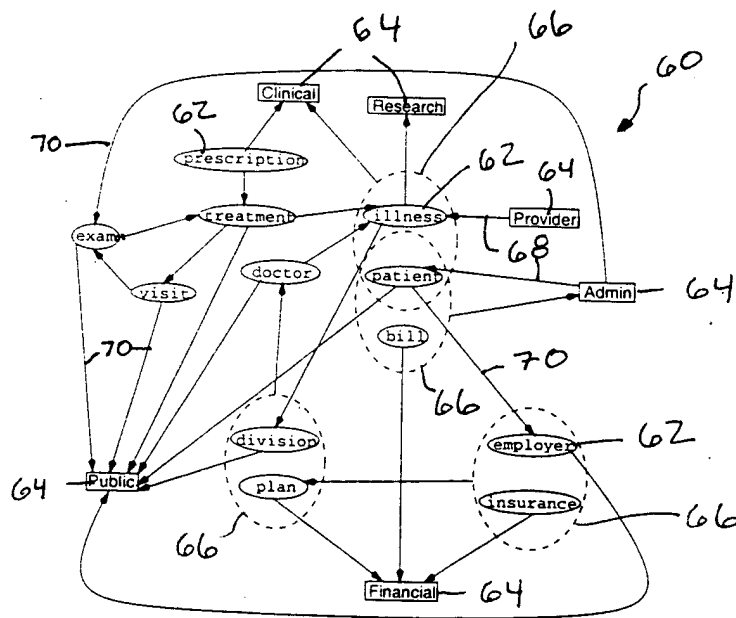


(a) A lattice diagram with nodes labeled with security levels and sets of categories. The top node is $\langle TS, \{Army, Nuclear\} \rangle$. Below it are three nodes: $\langle TS, \{Army\} \rangle$, $\langle S, \{Army, Nuclear\} \rangle$, and $\langle TS, \{Nuclear\} \rangle$. Below those are three nodes: $\langle S, \{Army\} \rangle$, $\langle TS, \{ \} \rangle$, and $\langle S, \{Nuclear\} \rangle$. The bottom node is $\langle S, \{ \} \rangle$. A handwritten '30' with an arrow points to the top node.

(b) A hierarchy diagram showing a vertical chain of security levels: TopSecret, Secret, Confidential, and Unclassified. A handwritten '40' with an arrow points to the TopSecret node.

(c) A diamond-shaped lattice diagram. The top node is HMO. Below it are Admin and Provider. Below Admin is Financial. Below Provider is Clinical. Below Clinical is Research. The bottom node is Public. A handwritten '50' with an arrow points to the HMO node.



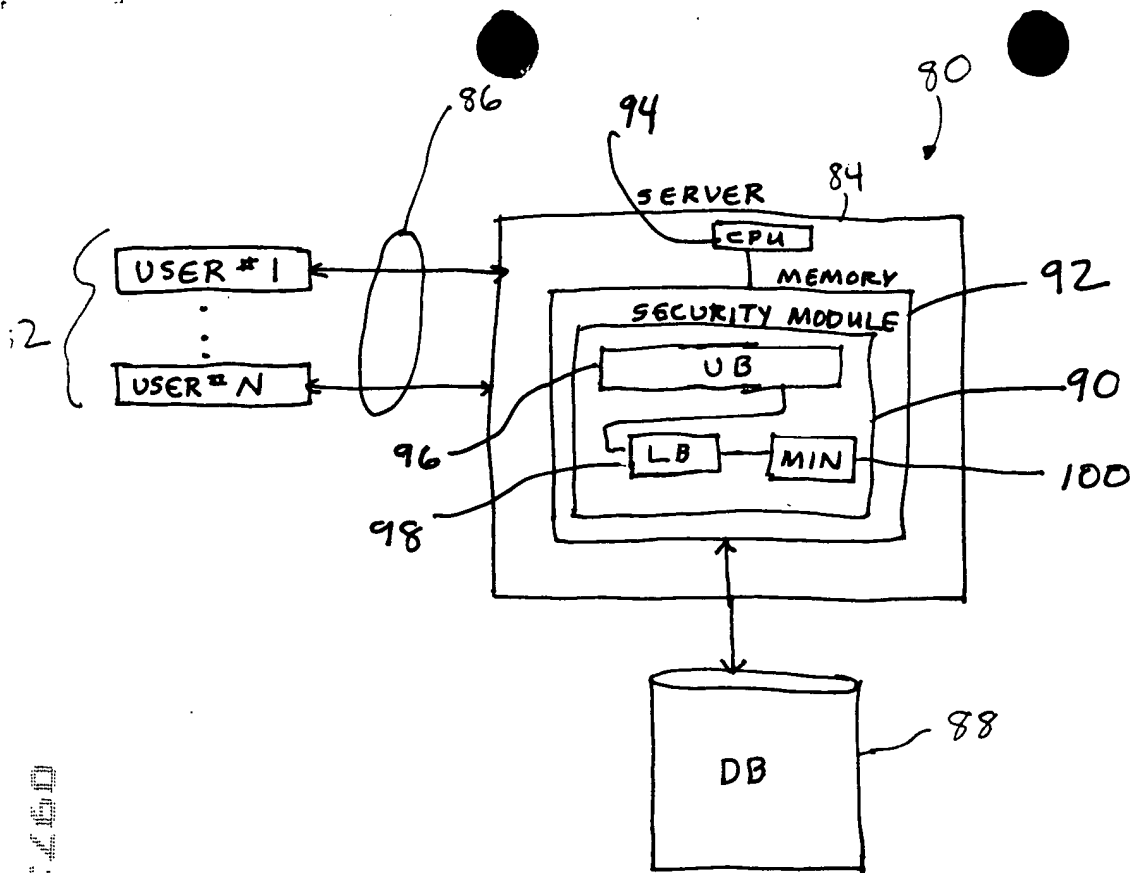


FIGURE 3

$\text{Provider} \succeq \lambda(\text{illness})$
 $\lambda(\text{illness}) \succeq \lambda(\text{division})$
 $\text{lub}\{\lambda(\text{division}), \lambda(\text{plan})\} \succeq \lambda(\text{doctor})$
 $\lambda(\text{illness}) \succeq \text{Research}$
 $\lambda(\text{division}) \succeq \text{Public}$
 $\lambda(\text{plan}) \succeq \text{Financial}$
 $\lambda(\text{doctor}) \succeq \text{Public}$

Figure 5(a)

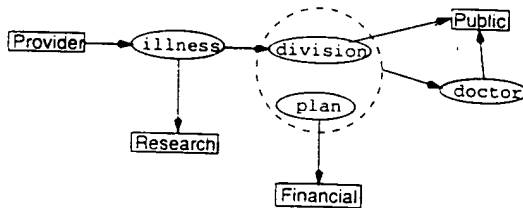


Figure 5(b)

$\lambda(\text{illness}) = \text{Provider}$
 $\lambda(\text{division}) = \text{Provider}$
 $\lambda(\text{plan}) = \text{HMO}$
 $\lambda(\text{doctor}) = \text{HMO}$

Figure 5(c)

0971152.10900

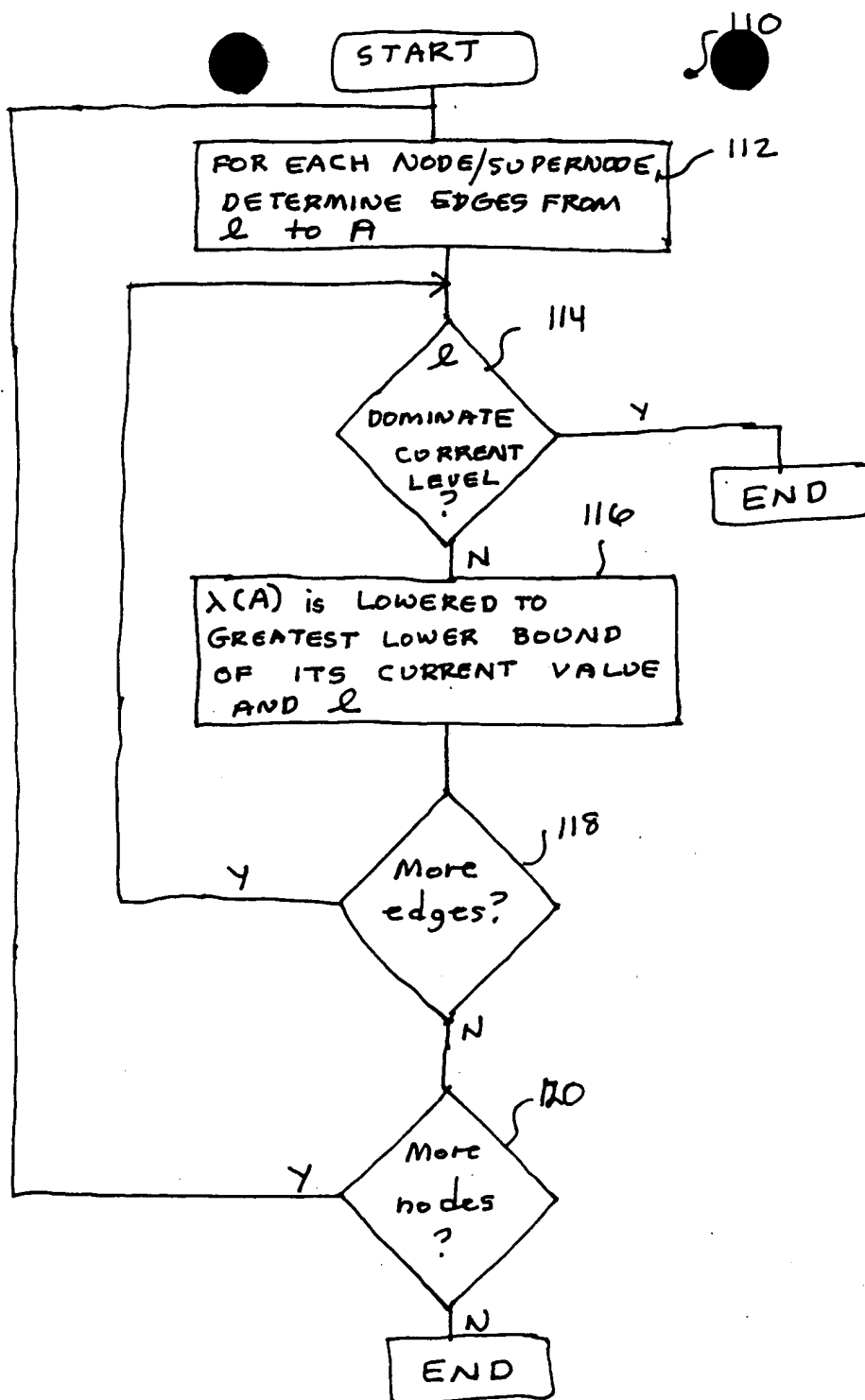


FIGURE 4

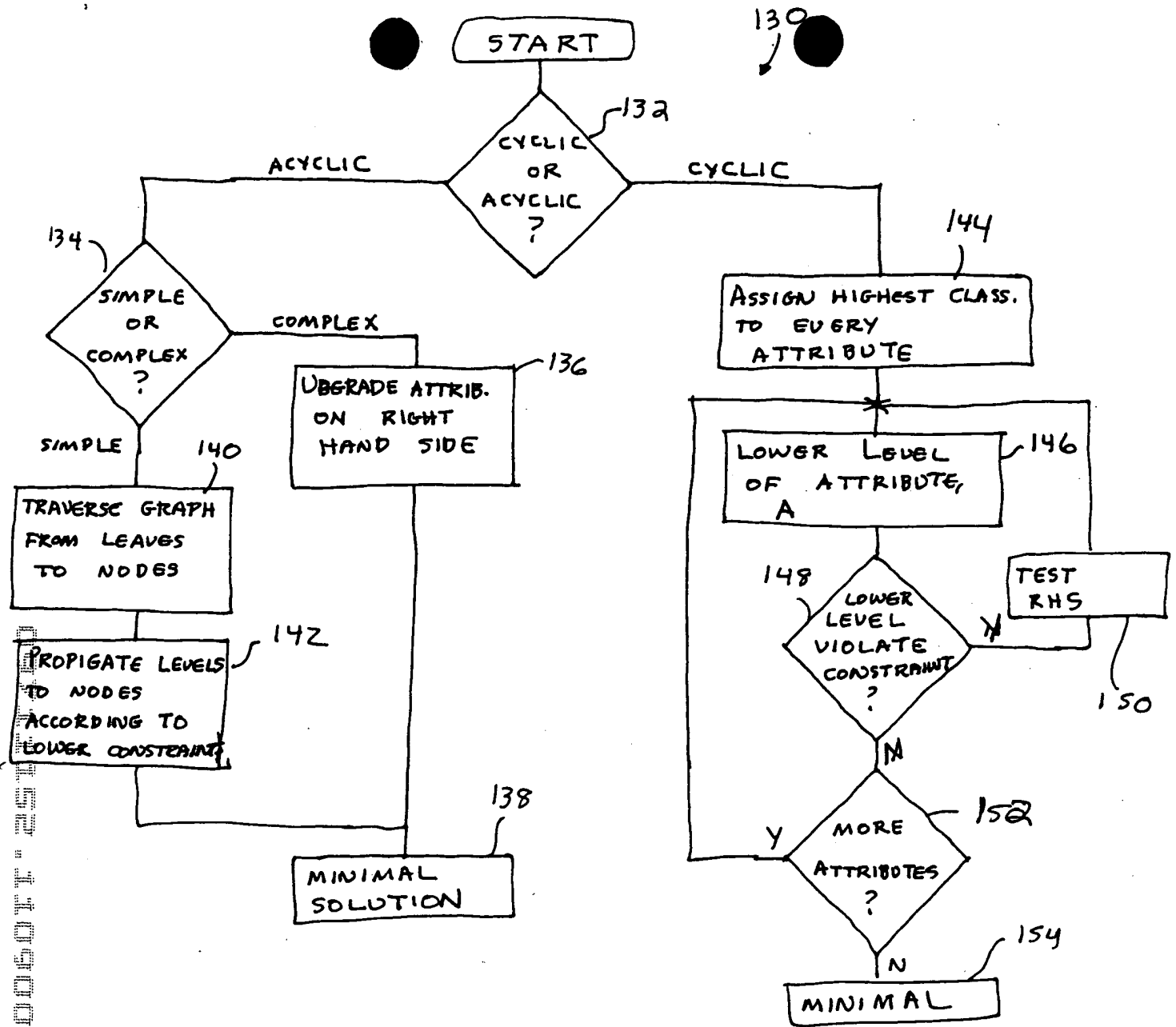
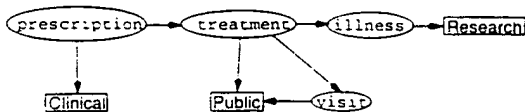


FIGURE 6

$\lambda(\text{illness}) \succeq \text{Research}$
 $\lambda(\text{prescription}) \succeq \text{Clinical}$
 $\lambda(\text{prescription}) \succeq \lambda(\text{treatment})$
 $\lambda(\text{treatment}) \succeq \text{Public}$
 $\lambda(\text{treatment}) \succeq \lambda(\text{visit})$
 $\lambda(\text{treatment}) \succeq \lambda(\text{illness})$
 $\lambda(\text{visit}) \succeq \text{Public}$



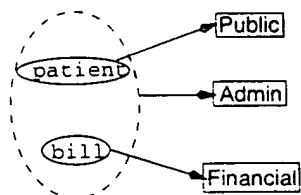
$\lambda(\text{illness}) = \text{Research}$
 $\lambda(\text{prescription}) = \text{Clinical}$
 $\lambda(\text{treatment}) = \text{Research}$
 $\lambda(\text{visit}) = \text{Public}$

Figure 7(a)

Figure 7(b)

Figure 7(c)

$\lambda(\text{patient}) \succeq \text{Public}$
 $\lambda(\text{bill}) \succeq \text{Financial}$
 $\text{lub}\{\lambda(\text{patient}), \lambda(\text{bill})\} \succeq \text{Admin}$



$\lambda(\text{bill}) = \text{Admin}$
 $\lambda(\text{patient}) = \text{Public}$

$\lambda(\text{bill}) = \text{Financial}$
 $\lambda(\text{patient}) = \text{Research}$

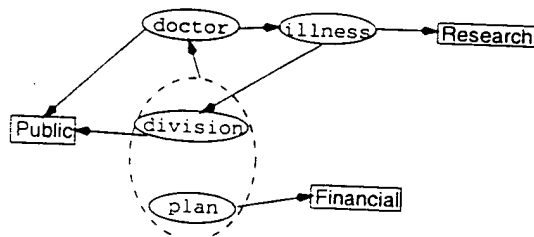
Figure 8(a)

Figure 8(b)

Figure 8(c)

Figure 8(d)

$\lambda(\text{division}) \succeq \text{Public}$
 $\text{lub}\{\lambda(\text{division}), \lambda(\text{plan})\} \succeq \lambda(\text{doctor})$
 $\lambda(\text{doctor}) \succeq \text{Public}$
 $\lambda(\text{doctor}) \succeq \lambda(\text{illness})$
 $\lambda(\text{illness}) \succeq \lambda(\text{division})$
 $\lambda(\text{illness}) \succeq \text{Research}$
 $\lambda(\text{plan}) \succeq \text{Financial}$



$\lambda(\text{division}) = \text{Public}$
 $\lambda(\text{doctor}) = \text{Research}$
 $\lambda(\text{illness}) = \text{Research}$
 $\lambda(\text{plan}) = \text{Admin}$

Figure 9(a)

Figure 9(b)

Figure 9(c)

005011-2511/60

Algorithm 3.1 (Minimal Classification Generation)

```

MAIN
For  $A \in \mathcal{A}$  do  $\text{Constr}[A] := \emptyset$ ;  $\text{visit}[A] := 0$ ;  $\text{done}[A] := \text{FALSE}$ 
For  $l \in L$  do  $\text{done}[l] := \text{TRUE}$ ;  $\text{visit}[l] := 1$ 
For  $c = (lhs, rhs) \in C_{lower}$  do
   $\text{count}[c] := 0$ 
  For  $A \in lhs$  do
     $\text{Constr}[A] := \text{Constr}[A] \cup \{c\}$ ;  $\text{count}[c] := \text{count}[c] + 1$ 
   $\text{Stack} := \emptyset$ 
  For  $A \in \mathcal{A}$  do
    If  $\text{visit}[A] = 0$  then  $\text{dfs\_visit}(A)$ 
   $\text{max\_scc} := 0$ 
  For  $i = 1, \dots, |\mathcal{A}|$  do  $\text{scc}[i] := ()$ 
  For  $A \in \mathcal{A}$  do  $\text{visit}[A] := 0$ 
  While NOTEMPTY( $\text{Stack}$ ) do
     $A := \text{POP}(\text{Stack})$ 
    If  $\text{visit}[A] = 0$  then
       $\text{max\_scc} := \text{max\_scc} + 1$ 
       $\text{scc}[\text{max\_scc}] := (A)$ 
       $\text{dfs\_back\_visit}(A)$ 
  For  $A \in \mathcal{A}$  do  $\lambda(A) := \top$ ;  $\text{visit}[A] := 0$ 
  compute_upper_bounds
  compute_partial_lubs
  compute_minimal_solution

COMPUTE_UPPER_BOUNDS
For  $(l, A) \in C_{upper}$  do  $\lambda(A) := \lambda(A) \sqcap l$ 
For  $i = 1, \dots, \text{max\_scc}$  do
  For  $A \in \text{scc}[i]$  do
    If  $\text{visit}[A] = 0$  then  $\text{upper\_bound}(A, i)$ 

UPPER_BOUND( $A, i$ )
 $\text{visit}[A] := 1$ 
For  $c = (lhs, rhs) \in \text{Constr}[A]$  do
  If  $\text{count}[c] > 0$  then  $\text{count}[c] := \text{count}[c] - 1$ 
  If  $\text{count}[c] = 0$  or  $rhs \in \text{scc}[i]$  then
     $\text{levlhs} := \perp$ 
    For  $A' \in lhs$  do  $\text{levlhs} := \text{levlhs} \sqcup \lambda(A')$ 
    If  $\neg(\text{levlhs} \geq \lambda(rhs))$  then
      If  $rhs \in L$  then Fail
      else  $\lambda(rhs) := \lambda(rhs) \sqcap \text{levlhs}$ 
      If  $rhs \in \text{scc}[i]$  then
         $\text{upper\_bound}(rhs, i)$ 

COMPUTE_MINIMAL_SOLUTION
For  $i := \text{max\_scc}, \dots, 1$  do
  For  $A \in \text{scc}[i]$  do
     $\text{done}[A] := \text{TRUE}$ ;  $l := \perp$ 
    For  $c = (lhs, rhs) \in \text{Constr}[A]$  do
      If  $\text{done}[rhs]$  then
        case  $|lhs|$  of
          1:  $l := l \sqcup \lambda(rhs)$ 
          >1:  $l := l \sqcup \text{minlevel}(A, c)$ 
      else  $\text{done}[A] := \text{FALSE}$ 
    If  $\text{done}[A]$  then  $\lambda(A) := l$ 
    else  $DSet := \{l' \mid l' \text{ is a maximal level, } \lambda(A) > l' \geq l\}$ 
    While  $DSet \neq \emptyset$ 
      Choose  $l''$  in  $DSet$ ;  $DSet := DSet - l''$ 
       $\text{Lower} := \text{try\_to\_lower}(A, l'')$ 
      If  $\text{Lower} \neq \emptyset$  then
        For  $(A', l') \in \text{Lower}$  do  $\lambda(A') := l'$ 
         $DSet := \{l' \mid l' \text{ maximal level, } \lambda(A) > l' \geq l\}$ 
     $\text{done}[A] := \text{TRUE}$ 
  For  $c \in \text{Constr}[A]$  do
     $j := \text{count}[c]$ ;  $\text{Plub}[c][j] := \lambda(A) \sqcup \text{Plub}[c][j+1]$ 
     $\text{count}[c] := \text{count}[c] - 1$ 

DFS_VISIT( $A$ )
 $\text{visit}[A] := 1$ 
For  $(lhs, rhs) \in \text{Constr}[A]$  do
  If  $\text{visit}[rhs] = 0$  then  $\text{dfs\_visit}(rhs)$ 
  PUSH( $A, \text{Stack}$ )

DFS_BACK_VISIT( $A$ )
/* Traverses the constraints backward and inserts all
attributes found in the same SCC list as  $A$  */
 $\text{visit}[A] := 1$ 
For  $(lhs, A) \in C_{lower}$  do
  For  $A' \in lhs$  do
    If  $\text{visit}[A'] = 0$  then
       $\text{scc}[\text{max\_scc}] := \text{concat}((A'), \text{scc}[\text{max\_scc}])$ 
       $\text{dfs\_back\_visit}(A')$ 

COMPUTE_PARTIAL_LUBS
For  $c = (lhs, rhs) \in C_{lower}$  do  $\text{count}[c] := 0$ ;  $\text{Plub}[c][0] := \perp$ 
For  $i = 1, \dots, \text{max\_scc}$  do
  For  $A \in \text{reverse}(\text{scc}[i])$  do
    For  $c = (lhs, rhs) \in \text{Constr}[A]$  do
       $\text{count}[c] := \text{count}[c] + 1$ ;  $j := \text{count}[c]$ 
       $\text{Plub}[c][j] := \text{Plub}[c][j-1] \sqcup \lambda(A)$ 
  For  $c = (lhs, rhs) \in C_{lower}$  do  $j := \text{count}[c] + 1$ ;  $\text{Plub}[c][j] := \perp$ 

MINLEVEL( $A, c$ )
/* Returns a minimal level for  $A$  that keeps  $c$  satisfied */
 $j := \text{count}[c]$ ;  $(lhs, rhs) := c$ ;  $\text{last} := \lambda(A)$ 
 $\text{lubothers} := \text{Plub}[c][j-1] \sqcup \text{Plub}[c][j+1]$ 
If  $\text{lubothers} \geq \lambda(rhs)$  then  $\text{last} := \perp$ 
else Try :=  $\{l \mid l \text{ is a maximal level s. t. } \text{last} > l\}$ 
While Try  $\neq \emptyset$  do
  Choose  $l$  in Try; Try := Try -  $l$ 
  if  $(l \sqcup \text{lubothers}) \geq \lambda(rhs)$  then
     $\text{last} := l$ ; Try :=  $\{l \mid l \text{ is a maximal level s. t. } \text{last} > l\}$ 
return last

TRY_TO_LOWER( $A, l$ )
Tocheck :=  $\{(A, l)\}$ 
Tolower :=  $\emptyset$ 
Repeat
  Choose  $(A', l') \in \text{Tocheck}$ 
  Tocheck := Tocheck -  $\{(A', l')\}$ 
  Tolower := Tolower  $\cup \{(A', l')\}$ 
  For  $(lhs, rhs) \in \text{Constr}[A']$  do
     $\text{level} := \perp$ 
    For  $A'' \in lhs$  do
      If  $\exists (A'', l'') \in \text{Tolower}$  then
         $\text{level} := \text{level} \sqcup l''$ 
      else  $\text{level} := \text{level} \sqcup \lambda(A'')$ 
    case  $\text{done}[rhs]$  of
      TRUE: If  $\neg(\text{level} \geq \lambda(rhs))$  then return  $\emptyset$ 
      FALSE: If  $\neg(\text{level} \geq \lambda(rhs))$  then
         $\text{newlevel} := \lambda(rhs) \sqcap \text{level}$ 
        If  $\exists (rhs, l'') \in (\text{Tolower} \cup \text{Tocheck})$  then
          If  $\neg(\text{newlevel} \geq l'')$  then
             $\text{newlevel} := l'' \sqcap \text{newlevel}$ 
          If  $(rhs, l'') \in \text{Tolower}$  then
            Tolower := Tolower -  $\{(rhs, l'')\}$ 
          else Tocheck := Tocheck -  $\{(rhs, l'')\}$ 
          Tocheck := Tocheck  $\cup \{(rhs, \text{newlevel})\}$ 
        else Tocheck := Tocheck  $\cup \{(rhs, \text{newlevel})\}$ 
until Tocheck =  $\emptyset$ 
return Tolower

```

Figure 10 Algorithm for computing a minimal classification.

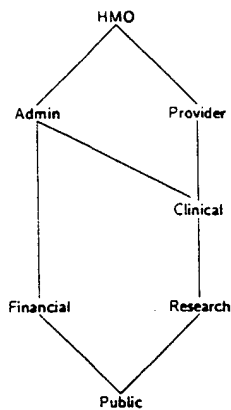


Figure 11 (a)

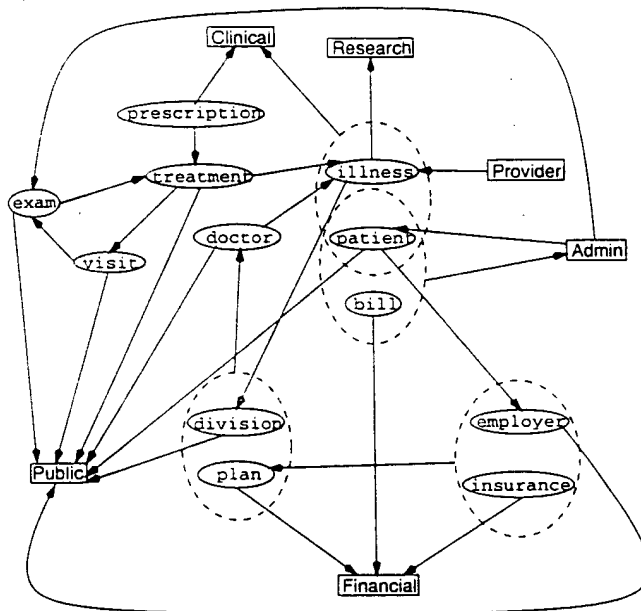


Figure 11 (b)

		SCC											
		[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]				
		doctor	division	illness	plan	employer	patient	bill	insurance	exam	treatment	visit	prescription
doctor	compute_upper_bounds	HMO	HMO	HMO	HMO	HMO	HMO	HMO	HMO	HMO	HMO	HMO	HMO
	try_to_lower(doctor, Admin) F	Admin	Clinical	Clinical	HMO	Admin	Admin	HMO	HMO	Admin	Admin	Admin	HMO
	try_to_lower(doctor, Financial) F	Admin	Clinical	Clinical	HMO	Admin	Admin	HMO	HMO	Admin	Admin	Admin	HMO
	try_to_lower(doctor, Clinical)	Clinical	Clinical	Clinical	HMO	Admin	Admin	HMO	HMO	Admin	Admin	Admin	HMO
	try_to_lower(doctor, Research)	Research	Research	Research	HMO	Admin	Admin	HMO	HMO	Admin	Admin	Admin	HMO
division	try_to_lower(doctor, Public) F	Research	Research	Research	HMO	Admin	Admin	HMO	HMO	Admin	Admin	Admin	HMO
	-		Public	Research	HMO	Admin	Admin	HMO	HMO	Admin	Admin	Admin	HMO
	-			Research	HMO	Admin	Admin	HMO	HMO	Admin	Admin	Admin	HMO
	-			Research	HMO	Admin	Admin	HMO	HMO	Admin	Admin	Admin	HMO
	-			Research	Admin	Admin	Public	HMO	HMO	Admin	Admin	Admin	HMO
illness	-				Admin	Admin	Clinical	Financial	Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
plan	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
employer	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
patient	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
bill	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
insurance	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
	-					Admin			Admin	Admin	Admin	Admin	HMO
exam	try_to_lower(exam, Financial) F									Admin	Admin	Admin	HMO
	try_to_lower(exam, Clinical)									Clinical	Clinical	Clinical	HMO
	try_to_lower(exam, Research)									Research	Research	Research	HMO
	try_to_lower(exam, Public) F									Research	Research	Research	HMO
	-									Research	Research	Research	HMO
treatment	-									Research	Research	Research	HMO
	-									Research	Research	Research	HMO
	-									Research	Research	Research	HMO
	-									Research	Research	Research	HMO
	-									Research	Research	Research	HMO
visit	-									Research	Research	Research	HMO
	-									Research	Research	Research	HMO
	-									Research	Research	Research	HMO
	-									Research	Research	Research	HMO
	-									Research	Research	Research	HMO
prescription	-									Research	Research	Research	HMO
	-									Research	Research	Research	HMO
	-									Research	Research	Research	HMO
	-									Research	Research	Research	HMO
	-									Research	Research	Research	HMO
• final levels		Research	Public	Research	Admin	Public	Clinical	Financial	Admin	Research	Research	Research	Clinical

Figure 11 (c)

SGC

		[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]				
* initial levels		doctor	division	illness	plan	employer	patient	bill	insurance	exam	treatment	visit	prescription
patient	compute_upper_bounds	HMO	HMO	HMO	HMO	Admin	Admin	HMO	HMO	Admin	Admin	Admin	HMO
	try_to_lower(patient, Financial)	HMO	Clinical	Clinical	HMO	Financial	Financial	HMO	HMO	Admin	Admin	Admin	HMO
	try_to_lower(patient, Public)	HMO	Clinical	Clinical	HMO	Public	Public	HMO	HMO	Admin	Admin	Admin	HMO
	try_to_lower(patient, Admin)	Admin	Clinical	Clinical	Admin	Public	Public	HMO	HMO	Admin	Admin	Admin	HMO
plan	try_to_lower(plan, Financial)	Admin	Clinical	Clinical	Financial	Public	Public	HMO	HMO	Admin	Admin	Admin	HMO
	try_to_lower(plan, Public)	Admin	Clinical	Clinical	Financial	Public	Public	HMO	HMO	Admin	Admin	Admin	HMO
doctor	try_to_lower(doctor, Financial)	Admin	Clinical	Clinical	Financial	Public	Public	HMO	HMO	Admin	Admin	Admin	HMO
	try_to_lower(doctor, Clinical)	Clinical	Clinical	Clinical	Financial	Public	Public	HMO	HMO	Admin	Admin	Admin	HMO
	try_to_lower(doctor, Research)	Clinical	Clinical	Clinical	Financial	Public	Public	HMO	HMO	Admin	Admin	Admin	HMO
	try_to_lower(doctor, Admin)	Clinical	Clinical	Clinical	Financial	Public	Public	HMO	HMO	Admin	Admin	Admin	HMO
division	-	Clinical	Research	Clinical	Financial	Public	Public	HMO	HMO	Admin	Admin	Admin	HMO
illness	-				Financial	Public	Public	HMO	HMO	Admin	Admin	Admin	HMO
employer	-					Public		HMO	HMO	Admin	Admin	Admin	HMO
bill	-							Admin	HMO	Admin	Admin	Admin	HMO
insurance	-								Financial	Admin	Admin	Admin	HMO
exam	try_to_lower(exam, Financial)									Admin	Admin	Admin	HMO
	try_to_lower(exam, Clinical)									Clinical	Clinical	Clinical	HMO
	try_to_lower(exam, Research)									Clinical	Clinical	Clinical	HMO
treatment	try_to_lower(treatment, Clinical)									Clinical	Clinical	Clinical	HMO
visit	-											Clinical	HMO
prescription	-												Clinical
* final levels		Clinical	Research	Clinical	Financial	Public	Public	Admin	Financial	Clinical	Clinical	Clinical	Clinical

Figure 12

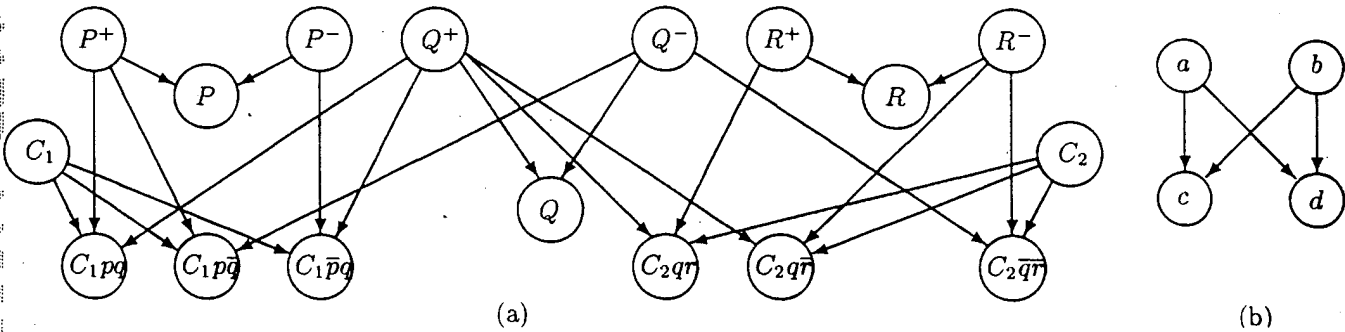


Figure 13 A + B

0060TT-25TT460